Teaching computer hardware and organisation using PIC-based projects

Nurul I. Sarkar
School of Computer & Information Sciences
Auckland University of Technology
Auckland, New Zealand
nurul.sarkar@aut.ac.nz

Trevor M. Craig
Wollongong University College
Auckland
New Zealand
Trevor.craig@wuc.ac.nz

Abstract

Until recently, it has been difficult to motivate students to learn about computer hardware because students find the subject rather dry and theoretical. We have prepared a series of interesting projects that give students a hands-on introduction to computer hardware and organisation. Our projects are designed around the PIC16F84, a powerful 8-bit microcontroller chip that sells for less than $10. The projects are suitable for classroom use in introductory-level courses about computer hardware. The effectiveness of these projects has been evaluated both formally by students (student-evaluation forms) and informally through discussion within the teaching team. The feedback from students indicates that the development and the implementation of the projects were successful. This paper describes the details of the projects completed to date, their overall effectiveness and our plans for further projects.

Keywords: computer hardware, hands-on experience, PIC microcontroller, evaluation

Introduction

Computer hardware, digital systems design, number systems, CPU, memory and I/O (input/output) ports are topics often included in computer science, electronics, and engineering courses as fundamental concepts involved in computer hardware and organisation. It is believed that incorporating practical demonstrations into these courses, to illustrate theoretical concepts, and providing opportunity for hands-on experience significantly enhances student learning about computer hardware. Yet very little material designed to supplement the teaching of computer hardware concepts is publicly available, as a search of the Computer Science Teaching Center [1] and SIGCSE Education Links [2] sites reveals.

We strongly believe, as do many others [3-5], that students learn more effectively from courses that provide for active involvement in hands-on activities. Towards that end, we have prepared some interesting and fun projects that facilitate an interactive, hands-on experience to computer hardware concepts. These projects are designed around the PIC16F84 (PIC stands for programmable interface controller) microcontroller chip (8-bit 4MHz), and can be used either in the classroom, as a demonstration, to enhance the traditional lecture environment, or in the laboratory to provide practical hands-on experience at an introductory level.

Computer hardware and organisation is described in many textbooks e.g. [4, 6, 7], and the PIC microcontroller is discussed extensively in the computer hardware literature e.g. [8-11]. [12] describes several low-cost PIC programming kits designed for the PIC16F84 chip, and [13] deals with MS Windows 2000 and XP-based PIC programming software for the PIC programmer and checkerboard. Assembled modules, such as those described in [14], which incorporate one or more PIC microcontrollers, are available commercially.

A number of sophisticated simulators exist for building a variety of computer architecture models [15-17]. Nevertheless, practical design and setting up demonstration projects the students gain first hand experience that cannot be gained through computer simulation and modeling. To date, we have focussed on developing some demonstration projects to support teaching traditional computer hardware and organisation. These projects are described in some detail along with our plans for future projects.

The remainder of this paper is organised as follows. We first discussed the PIC microcontroller environment. We then described a series of interesting projects completed to date and plans for future projects. The effectiveness of the PIC-based projects is evaluated and interpreted, and the main benefits of PIC projects are highlighted. Finally, we discussed the implementation of PIC projects in a laboratory settings followed by conclusion.

PIC Environment

Hardware

Microcontrollers contain the essential features of a computer on a single silicon chip – CPU, memory, I/O ports, timing and control circuitry. They are frequently found in roles befitting their name viz. controlling other devices such as VCR or DVD players and in motor vehicles, trains and planes – where the tasks at hand require computer control but do not require an extensive memory capacity. Some microcontrollers have additional circuitry such as an analogue-to-digital (A/D) converter built in, which makes them particularly useful in certain applications [9, 10]. The electrically erasable programmable read only memory (EEPROM) built into the PIC16F84 is particularly useful in the classroom as it enables the PIC to be reprogrammed many times over as students develop new programs. The computer-in-miniature aspect of microcontrollers makes them an ideal aid for teaching and learning about computer hardware and software principles. For example, the Harvard architecture of the PIC16F84 illustrates a computer architecture different from the von Neuman architecture commonly found in larger computers.

Software

Once a program has been written for a microcontroller, the program has to be entered into the memory of the microcontroller. With the PIC16F84 this can be achieved using the programmer module shown in Fig. 2. Details on the availability of this module and the associated software can be found in [8]. The software supplied with the programmer module enables programs written in BASIC on a PC (personal computer) to be quickly transferred to the PIC microcontroller. Reference [8] provides a step-by-step guide to creating programs for the PIC that become more advanced as the student's experience grows. The short turn-around time between program writing, installing the program in the EEPROM and testing the PIC in the circuitry developed on the breadboard makes this amalgam of hardware and software particularly effective as a learning tool.

Project details

Table 1 lists the five projects that have been developed to date and which were trialed over a period of one semester in our diploma in information technology course IT Concepts and Skills. This course is at level 4 or first year diploma in information technology at the Auckland University of Technology (AUT).

TABLE 1. PIC-Based Projects and Related Hardware Concepts

Projects	Hardware concepts
1. Data representation	Binary sequence, bits, nibbles, bytes
2. Memory	Memory addressing, flash memory, RAM, ROM, EEPROM
3. LED matrix	Nibble, word, decoding, encoding, LED matrix, voltage and logic levels, switches, non-interlaced scanning.
4. LCD display	CPU and registers, I/O port, LCD display
5. Speech generation	Processor, serial-to-parallel shift registers, amplifier

Project 1: Data representation

This introductory project demonstrates to students the basic concepts of bits, bytes, and binary numbers. In lectures, the basic concepts of number systems are introduced, and students are shown how to convert from binary-to-hex and vice-versa. The teacher then demonstrates the project to the class to illustrate the theoretical concepts of bits, bytes, and of a binary sequence counter. Through active participation in the demonstration and hands-on activities students gain basic knowledge about the binary number system. Figure 1 shows the breadboard with eight LEDs (light emitting diodes) and the PIC programmed to output 10011011. Figure 2 shows the PIC programmer module that connects to the PC parallel port via a ribbon cable connected to the 25-pin D-socket located top right on the module. Figure 3 shows a ZIF (Zero Insertion Force) socket and associated cable that can be connected to the PIC programmer module via the socket located center left in Fig. 2. Using the ZIF socket, instead of the DIP socket provided on the programmer board (see Fig. 2), reduces the chance of pin damage to the PIC while it is being inserted in or removed from the programmer module.

Project 2: Memory

The basic concepts of memory addressing, random access memory (RAM), flash memory, read-only memory (ROM), and electrically erasable programmable ROM (EEPROM) are introduced during lectures. This project shows students how digital data can be read from and written to RAM. Students also gain hands-on experience in reading data from and writing data to EEPROM, whether this is located within the PIC16F84 microcontroller, or externally on the breadboard.

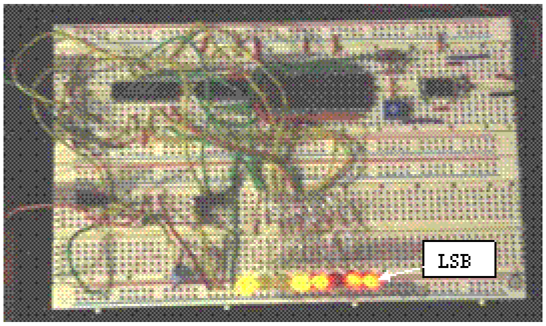

Fig.1. Project 1. The eight LED display represents the 1-byte word 10011011. The four yellow LEDs correspond to the most significant bits and the four red LEDs to the least significant bits.

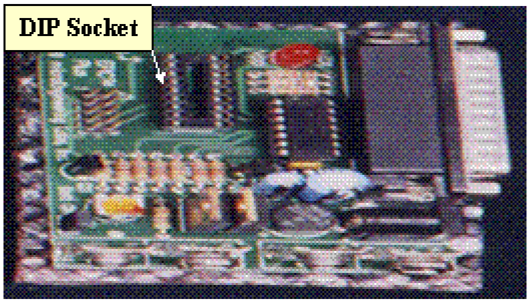

Fig. 2 PIC programmer module.

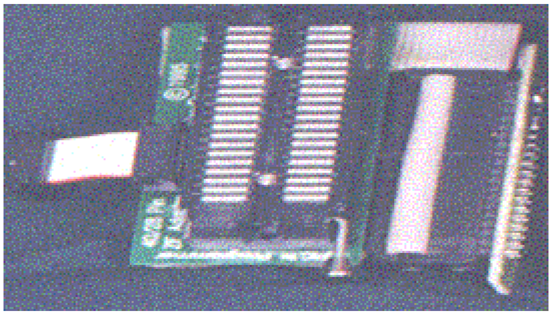

Fig. 3 ZIF (Zero Insertion Force) socket.

Project 3: LED matrix

Although, initially, the projects followed closely the projects described in [8], the students were encouraged to experiment once they had mastered the concepts in the original design. An example of this development is seen in Fig. 4, which shows a LED matrix module attached to the breadboard. The description of Project 3 that follows illustrates the many opportunities for turning principle into practice that PIC projects can present to students. In particular, students will learn about the way the 8 individual bits that make up a byte can be manipulated and utilized for purposes other than counting. The description of Project 3 and the project itself aptly illustrate the Chinese adage, attributed to Confucius (551-479 BC), ‘I hear, I know. I see, I remember. I do, I understand.’ As in Project 1, the PIC in Project 3 is programmed to produce 1-byte words. However, in addition to lighting the appropriate LEDs, which can be seen near the top left-hand corner of the breadboard, the

1-byte word in Project 3 is split into two 4-bit nibbles. These in turn are sent to two 74LS154 ICs (seen at the bottom of the matrix module), each of which activates 1 of its 16 output ports in accordance with the 4-bit input. Each of these output ports is linked to a column or a row of the LED matrix – in effect producing an (x, y) coordinate pair. Each point on the 16 x 16 LED matrix corresponds to an 8-bit word. By programming the PIC to produce the correct sequence of 8-bit words, a display such as that shown in Fig. 4 can be produced. This project also illustrates the distinction between logic levels and voltage levels. To turn an individual LED at (x, y) requires a positive voltage to be applied to row y, but the voltage level on a 74LS154 output port goes low when that port is selected. Consequently, four 74LS04 Hex Inverters were inserted between the 74LS154 connected to the 16 rows and the LEDs. These are seen on the left of the display.

If the matrix board is connected to the breadboard as shown in Figure 3 when the microcontroller is programmed with the binary counting program used in Project 1, a sequence of LEDs will light, starting at the bottom left-hand corner of the matrix (position (1,1)) and proceeding along each row in turn until the top or sixteenth row is reached (position (16,16)). At the end of the sixteenth row, the sequence repeats, starting at position (1,1). The students' attention can be drawn to the identical nature of this 'scanning' of the LED matrix and the scanning that occurs in a computer monitor, noting the rapid fly-back period when the 'spot' moves from the end of one line to the beginning of the next line (or the next scan). This analogous situation can be further exploited by asking the students how the displayed scanning procedure differs (if at all) from the nature of the scanning procedure in a television receiver. Students can be reminded that television uses inter-laced scanning in order to conserve bandwidth while simultaneously reducing flicker on the television screen to a level acceptable by viewers. A comprehension question for students is to ask them to replicate interlaced scanning of the LED matrix via either modifying the Binary Counting program in Project 1 or by modifying the hardware on the breadboard in some way, or perhaps by some method involving both the program and the hardware. If the bits in the byte produced by the Binary Counting program and displayed on the 8 LEDs, as shown in Fig. 1, are labeled b0 to b7, then bits b0 to b3 are used in Project 3 to control the x-coordinate on the LED matrix and bits b4 to b7 are used to control the y-coordinate. An illustration of the interplay between software and hardware solutions to problems is provided by simply changing the order in which the b4 to b7 bits are presented to the LED matrix board. This can be easily achieved on the breadboard by moving connector wires from b4, b5, b6, b7 to b7, b4, b5, b6. This action results in the odd lines (1, 3, 5, … 13, 15) of the LED matrix being scanned first followed by the even lines (2, 4, 6, … 14, 16) being scanned. The scan then starts again at line 1. The photograph in Fig. 4 has been retouched to improve the clarity of the LED matrix display.

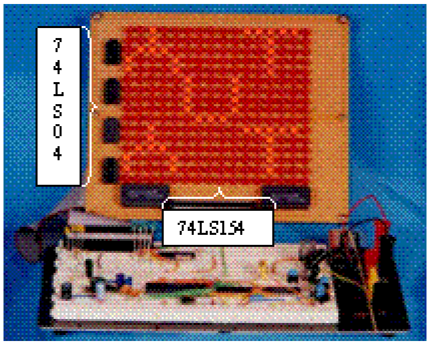

Fig. 4 Project 3. LED Matrix display.

Project 4: LCD display

This project utilizes a serial bit stream generated by the PIC 16F84 to create a two-line display on an LCD readout. The project follows the description in [8] but innovation on the part of the student is possible in the way that the PIC is programmed and in the content of the displayed message. This aspect of this project is enhanced when it is combined with the speech generator in Project 5. Fig. 5 shows the LCD displaying the message ‘Hi Nurul, how is life at the AUT?’

Project 5: Speech generation

This project is based around the SPO256 Speech Generator IC, which can produce 59 allophones. Although the speech produced does not sound completely natural, it is of sufficient quality to be easily comprehended. Reference [8] describes in detail the use of a serial to parallel converter (74LS164) to increase the number of I/O lines available from the PIC. This allows the PIC to control the speech generator (SPO256) so as to produce speech through the speaker shown in Fig. 5, and, at the same time, display on the LCD readout the words coming from the speaker. This combined project tests the student's mastery of the concepts (hardware and software) presented in the previous projects and many students appear to respond positively when their creation speaks and displays its 'first words'. By changing the clock frequency of the speech generator chip (by substituting a different crystal), or by using different resistors and/or capacitors to form the low pass filter that precedes the audio amplifier IC (LM836), students can gain first hand experience of the influence that changing the frequency content of an audio signal has upon the sound perceived by the listener. Some students found that, initially, they were unable to obtain any sound from this project and that the regulated voltage had dropped from 5 volts to 2 volts. Closer inspection revealed that the filter capacitor on the output of the 7805 regulator IC had been omitted. This resulted in a feedback loop being established and the resulting oscillations were the cause of the project failure. Provision of a 0.1 microfarad capacitor on the 5 volt output pin of the 7805 regulator solved this problem, and proved a useful learning experience for the students.

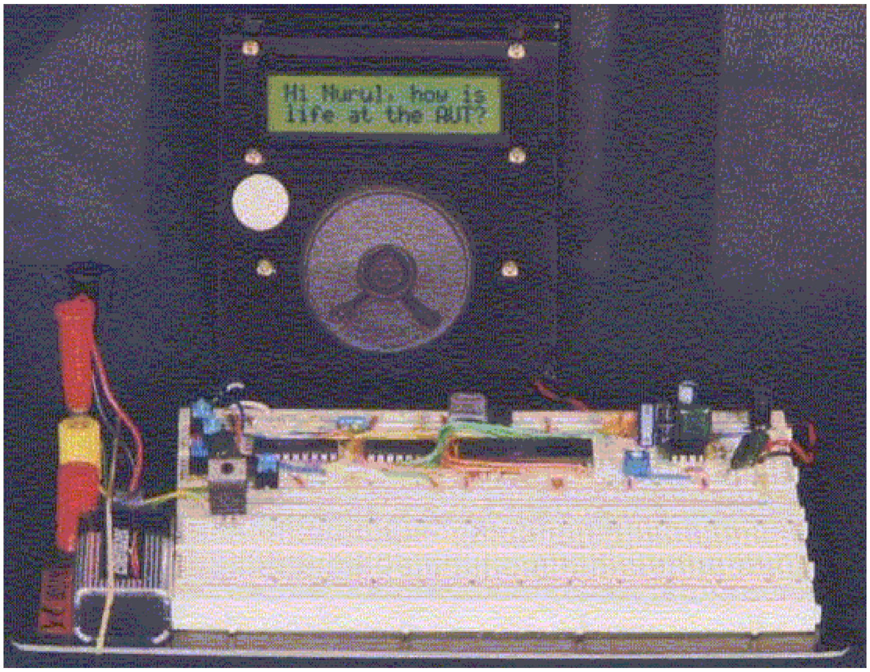

Fig. 5 LCD display (Project 4) combined with the speech generator project (Project 5).

Additional projects

The following projects are being considered:

- **Process control – DC motor control**: The hands-on learning activities include real-time process control system.
- **Pulse-code modulation - Analogue-digital converter**: The learning activities include sampling, analogue-to-digital converters, serial-to-parallel conversion.
- **Wireless technology:** Integrating both Infrared data from a remote control and Wi-Fi radio signals into the PIC projects to incorporate some aspects of data communications.
- **Open -** A category for student-suggested projects.

Analysis and Interpretation

To assess the educational value of the projects described in this paper, students were asked to complete a questionnaire. They were asked the following five questions:

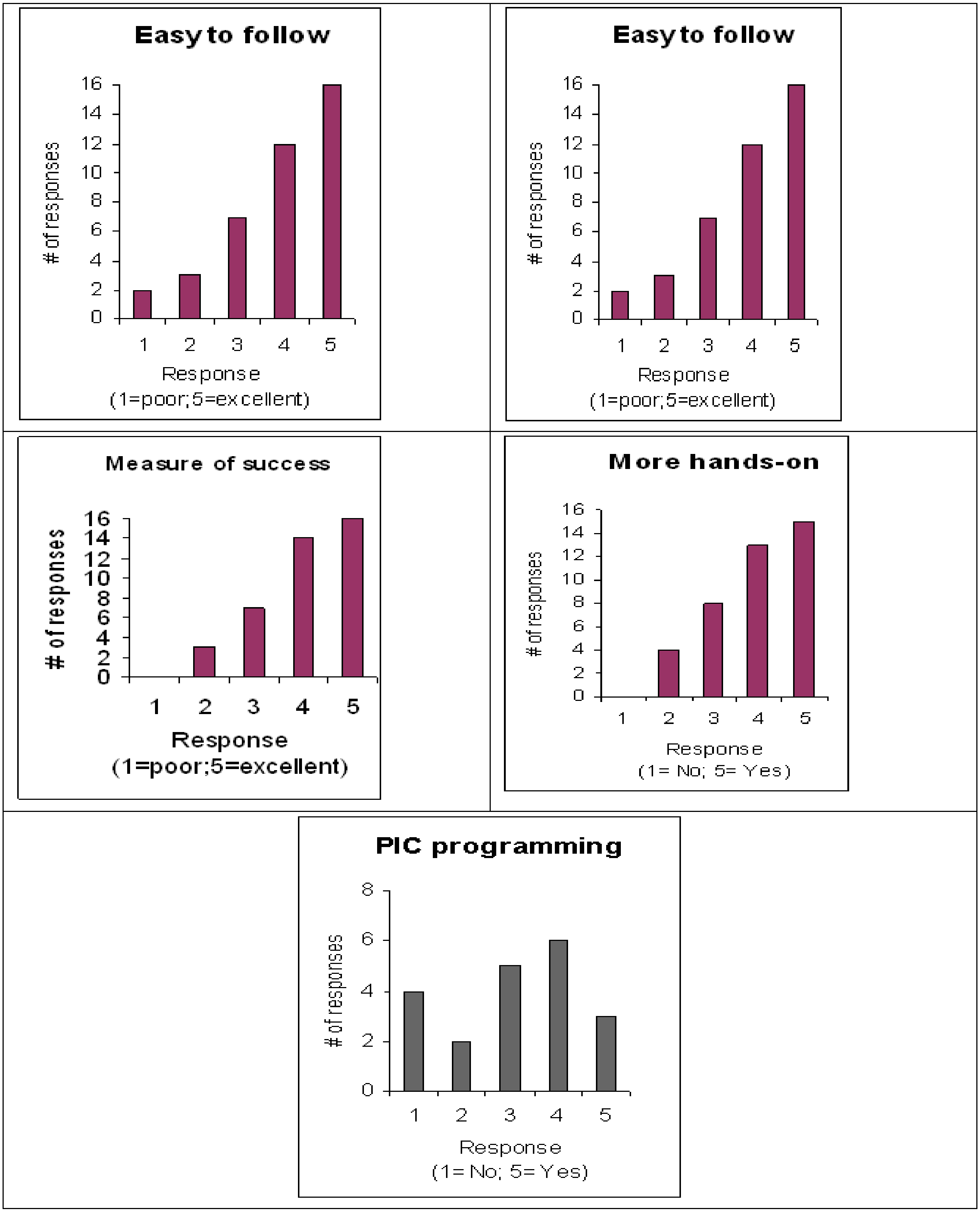

Fig. 6 Student-response graphs illustrating the number of respondents in each category for each of the five questions in the questionnaire.

1. **Prior knowledge:** How well did you understand computer hardware fundamentals before entering this course?
2. **Easy to follow:** How easy (overall) did you find the PIC-based projects to follow?
3. **Measure of success:** How effective were the PIC-based project demonstrations in helping you to improve your understanding of computer hardware concepts?
4. **Hands-on:** Would you like to have more projects of this kind as part of your course?
5. **PIC programming:** Would you prefer to learn to program the PIC microcontroller in BASIC or in another language such as Assembler or C?

A Likert scale with 5 points (1 to 5) was used in the Questionnaire. For Questions 1 to 3: 1= poor; 5 =excellent; for Questions 4 and 5: 1=No, 5 =Yes. Forty students responded and their responses are plotted in Fig. 6. The responses were interpreted as follows:

1. About 65% of the students indicated that they had no prior knowledge of computer hardware before entering this course. About 15% of the students had a basic knowledge of hardware whereas the rest (20%) were neutral (see Fig 6a).
2. The PIC-projects were found to be reasonably easy to use and set up for demonstrations. About 70% of the students were satisfied with the demonstrations and the hands-on experience, 12.5% expressed some concern and the rest (17.5%) were neutral (see Fig. 6b).
3. About 75% of the students indicated that the PIC-projects had clearly assisted them in developing a better understanding of hardware fundamentals. However, 7.5% of the students indicated that they were not totally satisfied with the PIC-projects and the rest (17.5%) were neutral (see Fig. 6c).

4. About 70% of the students indicated that they would like to have more hands-on projects in the course. About 10% were not very interested in trying more projects, and the rest (20%) were neutral (see Fig. 6d).
5. Most students were not very familiar with using programming languages (such as C or Assembler) to program microcontrollers, and appeared to be quite satisfied with using BASIC to program the PIC microcontroller (see Fig. 6e).

We observed that by participating in the PIC-based projects and demonstration activities, students became increasingly motivated to learn more about computer hardware and enjoyed this course more than previous courses that consisted of lectures only. We are seeking feedback regularly both from students and staff for further improvement of the demonstration materials.

Benefits

The PIC-based projects described in this paper provide the following benefits:

- **Low cost:** The current street price of a PIC16F84 is $7.5.
- **Easy to use:** PIC-based projects are easy to use and set up for demonstrations.
- **Hands-on:** PIC projects facilitate an interactive, hands-on introduction to computer hardware concepts.
- **Reusability:** Some hardware components of PIC projects can be reused in developing a variety of projects on a breadboard.
- **Programmable:** The PIC16F84 can be reprogrammed many times over as students develop new programs for a variety of projects.
- **Challenging:** The PIC projects provide a challenge to students' knowledge about computer hardware and software principles.

- **Usefulness:** The PIC projects reported in this paper can be used either in the classroom or in the laboratory to provide hands-on experience.

Implementation

Figure 7 illustrate how PIC projects can be setup and used in a typical computer laboratory to provide a hands-on experience in computer hardware and organization. A typical medium size computer laboratory with 25 computers, allows a class size of up to 25 students. As can be seen in Fig. 7, the PIC compiler and associated software is installed and run on a standalone PC (labeled Old PC), which is linked to a PIC programmer via a printer port (DB25-pin connector). The reason for using a stand alone PC is to have a single copy of the PIC compiler, which is more economical and eliminates the need for multiple licensing.

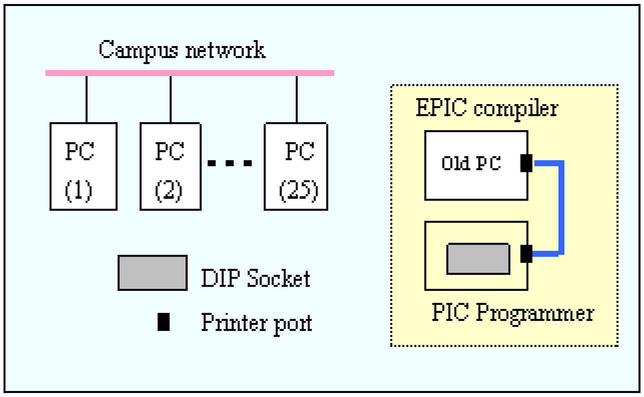

Fig. 7 PIC programmer set up in a typical laboratory.

Concluding remarks

A series of interesting projects has been developed, which can be used either in the classroom for class demonstrations, to enhance the traditional lecture environment, or in the computer laboratory for hands-on practical work at an introductory level. Student responses to the project demonstrations were mostly favorable and positive. The students indicated that they had found the PIC-based projects easy to use and helpful in gaining an understanding of a number of computer hardware concepts. More projects such as an analogue-to-digital converter, a dc motor controller and integrating both Infrared data from a remote control and Wi-Fi radio signals into the PIC projects are under development. Our materials, including the source code for all PIC applications are freely available to faculty interested in using the PIC projects to supplement their computer hardware courses or as the basis for more complex projects. More information about PIC-based projects and demonstration materials can be obtained by contacting the first author.

Acknowledgements

We would like to thank Nixon Lee and Koli Funaki for setting up the demonstrations.

References

[1] S. Grissom, D. Knox, E. Fox, and R. Heller, "Computer Science Teaching Center (CSTC). http://www.cstc.org/," 2005.

[2] Anonymous, "SIGCSE Education Links. http://sigcse.org/topics/," 2005.

[3] A. Leva, "A Hands-on Experimental Laboratory for Undergraduate Courses in Automatic Control," *IEEE Transactions on Education*, vol. 46, pp. 263-272, 2003.

[4] G. B. Shelly, T. J. Cashman, and M. E. Vermaat, *Discovering Computers 2004:Complete*, . Course Technology, 2003.

[5] R. D. Williams, R. H. Klenke, and J. H. Aylor, "Teaching Computer Design Using Virtual Prototyping.," *IEEE Transactions on Education*, vol. 46, pp. 296-301, 2003.

[6] I. Englander, *The Architecture of computer hardware and systems software - An information technology approach*, 2nd ed: John Wiley & Sons, Inc., 2000.

[7] A. S. Tanenbaum, *Structured computer organization*, 4th ed: Prentice Hall, Inc., 1999.

[8] J. Iovine, *PIC Microcontroller Project Book*: McGraw-Hill Companies, Inc., 2000.

[9] M. James, *Microcontroller Cookbook PIC & 8051*, 2nd ed: Newnes, 2001.

[10] J. Morton, *PIC Your Personal Introductory Course*, 2nd ed: Newnes, 2001.

[11] M. Predko, "Some useful book references on PIC Microcontrollers can be found at www.myke.com.," 2003.

[12] J. Rowe, "Three do-it yourself PIC programmers," in *Silicon Chip*, 2003, pp. 58-65.

[13] P. Smith, "Updating the PIC programmer and checkerboard.," in *Silicon Chip*, 2003, pp. 79-81.

[14] Comfile Technology, "Comfile Technology - http://www.comfile.co.kr/english2/study/pbstudy.html," 2003.

[15] E. Z. Bem and L. Petelczyc, "MiniMIPS - a Simulation Project for the Computer Architecture Laboratory.," presented at Proceedings of the 34th Technical Symposium on Computer Science Education (SIGCSE'03), Reno, Nevada, USA, 2003.

[16] R. N. Ibbett, "WWW Visualization of Computer Architecture Simulations.," presented at 7th annual SIGCSE conference on Innovation and Technology in Computer Science Education (ITiCSE), Aarhus, Denmark, 2002.

[17] B. Shelburne, "Teaching computer organization using a PDP-8 simulator," presented at SIGCSE'03 Technical Symposium on Computer Science Education, 2003.